

9/5/1 (Item 1 from file: 347)
DIALOG(R) File 347:JAPIO
(c) 2000 JPO & JAPIO. All rts. reserv.

04355212 **Image available**
DECENTRALIZED PROGRAM CONTROLLER

PUB. NO.: 05-346912 JP 5346912 A]
PUBLISHED: December 27, 1993 (19931227)
INVENTOR(s): AIURA TOSHIJI
APPLICANT(s): MITSUBISHI ELECTRIC CORP [000601] (A Japanese Company or
Corporation), JP (Japan)
APPL. NO.: 04-156586 [JP 92156586]
FILED: June 16, 1992 (19920616)
INTL CLASS: [5] G06F-015/16; G06F-015/16; G06F-011/28
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications); 45.1
(INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1721, Vol. 18, No. 189, Pg. 76, March
31, 1994 (19940331)

ABSTRACT

PURPOSE: To obtain a decentralized program debugging device with good operability which can send a debugging command to plural debuggers at a time by single input operation and makes the debuggers execute the command at the same time.

CONSTITUTION: A user inputs the debugging command together with identifiers given to the debuggers 5 and 7. A debugger controller 9 interprets the identifiers and sends the debugging command to local debugger controllers 10 and 11 which controls the debuggers 5 and 7 corresponding to the identifiers. The local debugger controllers 10 and 11 sends the debugging command to local debuggers 5 and 7, which executes the debugging command. Consequently, the user can send the command to the decentralized debuggers 5 and 7 by inputting the debugging command only once and the operability of decentralized program debugging is improved.

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平5-346912

(43)公開日 平成5年(1993)12月27日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	FI	技術表示箇所
G 0 6 F 15/16	4 5 0 Z	9190-5L		
	3 7 0 N	8840-5L		
11/28	J	9290-5B		

審査請求 未請求 請求項の数1(全 9 頁)

(21)出願番号 特願平4-156586

(22)出願日 平成4年(1992)6月16日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 相浦 利治

鎌倉市大船五丁目1番1号 三菱電機株式

会社情報電子研究所内

(74)代理人 弁理士 高田 守

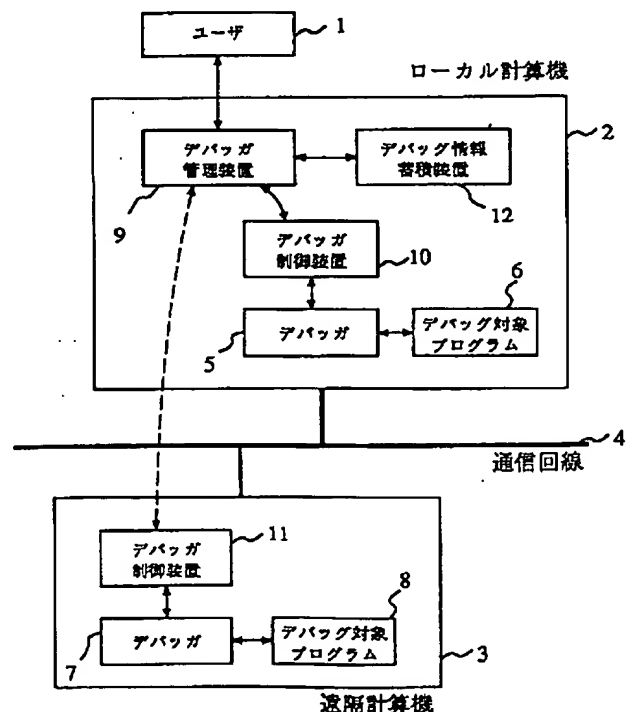
(54)【発明の名称】 分散プログラム制御装置

(57)【要約】

【目的】 1回の入力で同時に複数のデバッガに対してデバッグコマンドが送れ、そのコマンドが同時刻にデバッガに実行される操作性の良い分散プログラムデバッグ装置を得る。

【構成】 ユーザ1がデバッガ5、7に付けられた識別子と一緒にデバッグコマンドを入力する。デバッガ管理装置9が識別子を解釈し、識別子に対応するデバッガを制御しているローカルのデバッガ制御装置10、11へデバッグコマンドを送る。ローカルのデバッガ制御装置はデバッグコマンドをローカルのデバッガへ送り、デバッグコマンドが実行される。

【効果】 ユーザは1回のデバッグコマンド入力で、分散する複数のデバッガにコマンドを送ることができ、分散プログラムデバッグの操作性が向上する。



【特許請求の範囲】

【請求項1】 以下の要素を有する分散プログラム制御装置

(a) 分散して動作するプログラムに対応して設けられ、プログラムの動作を制御する制御手段、

(b) 上記制御手段を識別する識別子を付加したコマンドを入力し、識別子により識別され制御手段に対してコマンドを伝達する管理手段。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、例えば、計算機がネットワークで接続された分散環境において、遠隔計算機のプログラムと通信を行う分散プログラムのデバッグを行う装置に関するものである。

【0002】

【従来の技術】 従来技術の例として、UNIX System V (UNIXオペレーティングシステムはUNIX System Laboratories, Inc. が開発しライセンスしている。) プログラマ・リファレンス・マニュアルのSDB(1)に示されるシンボリック・デバッガがある。図6及び図7は、シンボリック・デバッガを使用して分散プログラムをデバッグする時の構成図を示している。

【0003】 図6において、1は分散プログラムのデバッグを行うユーザである。2はユーザ1の利用するローカル計算機であり、ユーザはローカル計算機2の端末装置を使用してデバッグを行う。3は遠隔のデバッグ対象プログラムの動作する遠隔計算機である。4はローカル計算機2と遠隔計算機3を接続する通信回線である。5はユーザ1が入力したデバッグコマンドをもとにローカルのデバッグ対象プログラム6を制御し、ローカルのデバッグ対象プログラム6の出力をユーザ1へ出力するローカルのデバッガである。6はローカル計算機2のローカルのデバッグ対象プログラムである。20はユーザの入力するデバッグコマンドを、通信回線4を介して遠隔のデバッガ7へ送信する、また遠隔のデバッガ7の出力を、通信回線4を介して受信しユーザへ出力する遠隔ログインである。7は遠隔ログイン20から送られたデバッグコマンドをもとに遠隔のデバッグ対象プログラム8を制御し、遠隔のデバッグ対象プログラム8の出力を遠隔ログイン20へ出力する遠隔のデバッガである。8は遠隔計算機3の遠隔のデバッグ対象プログラムである。

【0004】 ユーザ1は、図7のようなローカル計算機2に接続されたウィンドウ環境の、ローカルデバッガのウィンドウターミナル21、遠隔デバッガのウィンドウターミナル23、もしくはローカルデバッガのキャラクタ端末22、遠隔デバッガのキャラクタ端末24を用いて分散プログラムのデバッグを行う。まず、ローカルデバッガのウィンドウターミナル21もしくはローカルデバッガのキャラクタ端末22からローカルのデバッガ5を起動する。このときローカルのデバッグ対象プログラム

6を指定することで、ローカルのデバッグ対象プログラム6も同時に起動する。遠隔デバッガのウィンドウターミナル23もしくは遠隔デバッガのキャラクタ端末24から、遠隔ログイン20によって遠隔計算機3のデバッガ7を起動する。このとき遠隔のデバッグ対象プログラム8を指定することで、遠隔のデバッグ対象プログラム8も同時に起動する。

【0005】 ユーザ1がローカルのデバッグ対象プログラム6へデバッグコマンドを送りたい場合は、ローカルデバッガのウィンドウターミナル21もしくはローカルデバッガのキャラクタ端末22へコマンドを入力する。また、遠隔のデバッグ対象プログラム8へデバッグコマンドを送りたい場合は、遠隔デバッガのウィンドウターミナル23もしくは遠隔デバッガのキャラクタ端末24へコマンドを入力する。このようにすれば、ユーザは分散プログラムのデバッグが可能となる。また、ここではローカル計算機、遠隔計算機に各1つずつのデバッグ対象プログラムしかない場合についてのみ説明したが、ローカル計算機、遠隔計算機で複数のプログラムのデバッグを行うこともできる。

【0006】

【発明が解決しようとする課題】 従来のシンボリック・デバッガを用いた分散プログラムデバッグは以上のように構成されているので、ユーザがキャラクタ端末を使用していた場合、ローカルのデバッガ5が動作しているローカルデバッガのキャラクタ端末22を操作した後、遠隔のデバッガ7を操作するためには、ローカルデバッガのキャラクタ端末22のキーボードから手を離し、遠隔デバッガのキャラクタ端末24のキーボードから操作を行わなければならない。さらに多数の分散プログラムをデバッグ行う場合は、使用するキャラクタ端末の数も増え、キーボードの切替え作業も増加する。また、ユーザ1が、ウィンドウ環境を使用していた場合、ローカルデバッガのウィンドウターミナル21から遠隔デバッガのウィンドウターミナル23へウィンドウの切替を行わなければならない。このため、多数のプログラムをデバッグする場合には、ウィンドウ切替え作業の回数も増加する。以上の様な状況では、デバッガを切替えるための動作が、デバッグ処理の連続性を妨げているという点で操作性が悪い。また、分散プログラムのデバッグには、複数のプログラムから同時に1つのプログラムに対して通信をした場合の処理の振舞いを観測したいといった要求もある。この場合、複数のプログラムに同時にデバッグコマンドを送信しなければならないが、従来例においては不可能であった。

【0007】 この発明は、上記のような問題点を解消するためになされたもので、キャラクタ端末を使用していた場合にはデバッグを行なっている端末を離れることなく、また、ウィンドウ環境を使用していた場合にはウィンドウの切替を行わずに、ローカル計算機及び遠隔計

3

算機のデバッグにデバッグコマンドを送ることができるとともに、同時刻に複数のデバッグに対してデバッグコマンドを送ることができる分散プログラムデバッグ装置を得ることを目的とする。

【0008】

【課題を解決するための手段】この発明による分散プログラム制御装置は、例えば、デバッグ管理装置がデバッグ対象プログラムを起動する時に、デバッグ対象プログラム毎に異った識別子を付け、ユーザはデバッグコマンドを入力する時に、コマンドと送信先デバッグを示す識別子を入力するようにしたものであり、以下の要素を有するものである。

(a) 分散して動作するプログラムに対応して設けられ、プログラムの動作を制御する制御手段、(b) 上記制御手段を識別する識別子を付加したコマンドを入力し、識別子により識別され制御手段に対してコマンドを伝達する管理手段。

【0009】

【作用】この発明においては、管理手段は、ユーザから入力されたコマンドの識別子を解釈しローカル計算機もしくは遠隔計算機のプログラムにコマンドを分配配送する。よって、キャラクタ端末を使用しているユーザはキャラクタ端末間を移動することなく、また、ウィンドウターミナルを使用しているユーザはウィンドウを切替えることなく、分散しているプログラムにコマンドを送信することを可能とする。

【0010】

【実施例】実施例1. 図1は、発明の実施例1を示す図であり、1～8は、上記従来例と同様である。9は、ユーザ1から入力されるデバッグコマンドと送信先デバッグの識別子を解釈し、送信先デバッグを制御しているデバッグ制御装置10及び11へデバッグコマンドを出力するデバッグ管理装置である。10はデバッグ管理装置9から送られたデバッグコマンドを制御対象のローカルのデバッグ5へ出力し、ローカルのデバッグ5の出力をデバッグ管理装置9へ出力するローカルのデバッグ制御装置である。11はデバッグ管理装置9から送られたデバッグコマンドを制御対象の遠隔のデバッグ7へ出力し、遠隔のデバッグ7の出力をデバッグ管理装置9へ出力する遠隔のデバッグ制御装置である。12はローカルのデバッグ制御装置10、遠隔のデバッグ制御装置11を経て送られてくる各デバッグの出力をデバッグ毎に蓄積し、デバッグ管理装置9からの要求により指定されたデバッグの出力をデバッグ管理装置9へ出力するデバッグ情報蓄積装置である。

【0011】図2は、発明の実施例1をウィンドウ環境で動作させた場合の動作例を示す図である。15は、ローカルのデバッグ5の出力を表示するローカルのデバッグのウィンドウである。16は、遠隔のデバッグ7の出力を表示する遠隔デバッグのウィンドウである。17

4

は、デバッグ管理装置9へユーザが入力を行なう時に使用するコマンド入力ウィンドウである。

【0012】図3は、発明の実施例1で、ユーザ1がデバッグ管理装置9へ入力するデバッグコマンドを示したものである。(a)は、デバッグを起動する時に入力するコマンドである。ホスト名にデバッグ対象プログラムの動作する計算機名を与える。プログラム名にはデバッグ対象プログラムのプログラム名を与える。コマンド

(a)により、デバッグ対象プログラムのデバッグが可能となる。コマンド(a)を入力すると、動作している複数のデバッグを識別するための識別子が出力される。識別子は動作しているデバッグ毎に異った値が付けられる。以後、デバッグコマンドを入力する時には、識別子を付けることによってコマンドを送るデバッグを指定する。(b)は、デバッグに対してデバッグコマンドを送る時に入力するコマンドである。識別子1、識別子2にはデバッグ管理装置9が指定したデバッグの識別子を与える。このコマンドにより識別子1、識別子2に示されるデバッグに対してデバッグコマンドが送られる。

(c)は、デバッグの出力を表示させる時に入力するコマンドである。識別子1にはデバッグ管理装置9が指定したデバッグの識別子を与える。このコマンドにより、識別子1のデバッグの出力が表示される。(d)は、時刻指定でデバッグコマンドを実行させる時に入力するコマンドである。識別子1、識別子2にはデバッグ管理装置9が指定したデバッグの識別子を与える。このコマンドにより識別子1、識別子2に示されるデバッグは、12時23分15秒にデバッグコマンドを実行する。(e)は、相対時刻指定でデバッグコマンドを送る時に入力するコマンドである。識別子1、識別子2にはデバッグ管理装置9が指定したデバッグの識別子を与える。このコマンドにより識別子1に示されるデバッグは現在の時刻より20秒後に、また、識別子2に示されるデバッグは現在の時刻より30秒後に、デバッグコマンドを実行する。

【0013】次に上記実施例1の動作を図1～図3を参照しながら説明する。まず、ユーザ1がローカル計算機2で本装置を起動することにより、デバッグ管理装置9とデバッグ情報蓄積装置12が起動される。ユーザ1がローカルのデバッグ対象プログラム6のデバッグを示すコマンドを入力するとローカルのデバッグ制御装置10が起動し、ローカルのデバッグ制御装置10によってローカルのデバッグ5及びローカルのデバッグ対象プログラム6が起動される。ローカルのデバッグ対象プログラム6起動後に、デバッグ管理装置9がローカルのデバッグ対象プログラム6に対して識別子を付け表示する。次に、ユーザ1が遠隔のデバッグ対象プログラム8のデバッグを示すコマンドを入力すると遠隔のデバッグ制御装置11が起動し、遠隔のデバッグ制御装置11によって遠隔のデバッグ7及び遠隔のデバッグ対象プログラム8が起動される。遠隔のデバッグ対象プログラム8起動後

5

に、デバッグ管理装置9が遠隔のデバッグ対象プログラム8に対して識別子を付け表示する。

【0014】ユーザ1はデバッグ管理装置9へ入力するデバッグコマンドに識別子を付けることで、デバッグコマンドの送信先デバッグを指定する。ユーザ1が送り先をローカルのデバッグ5としてデバッグコマンド入力した場合、デバッグ管理装置9は、入力されたコマンドから送信先デバッグを解釈し、そのコマンドをローカルのデバッグ制御装置10へ送る。ローカルのデバッグ制御装置10はデバッグ管理装置9から送られたコマンドをローカルのデバッグ5へ出力することにより、ユーザ1のコマンドが送り先のローカルのデバッグ5へ送られる。また、ローカルのデバッグ制御装置10はローカルのデバッグ5の出力をデバッグ管理装置9へ送る。デバッグ管理装置9はローカルのデバッグ5の出力をデバッグ情報蓄積装置12へ送り、デバッグ情報蓄積装置12はそれをローカルのデバッグ5の出力として蓄積する。ユーザ1がローカルのデバッグ5の出力を表示させるコマンドをデバッグ管理装置9に入力すると、デバッグ管理装置9はデバッグ情報蓄積装置12よりローカルのデバッグ5の出力を獲得し表示する。ユーザ1が、デバッグコマンドの送信先を複数のデバッグに対して指定すると、デバッグ管理装置9がそれを解釈し、それぞれのデバッグ制御装置10へコマンドが送られる。これにより複数のデバッグに対して同時にデバッグコマンドを送ることが可能となる。

【0015】ユーザ1が、図2のようなウィンドウ環境でデバッグを行っていた場合は、本装置を起動することによりコマンド入力ウィンドウ17がディスプレイに表示される。以降、ユーザ1はコマンド入力ウィンドウ17からデバッグ管理装置9に対してデバッグコマンドを送る。ユーザ1が、デバッグ管理装置9にデバッグ5の出力を表示するコマンドを送ると、ウィンドウターミナル15がディスプレイに表示され、デバッグ5の出力はウィンドウターミナル15に出力される。同様にして、ユーザ1が、デバッグ管理装置9にデバッグ7の出力を表示するコマンドを送ると、ウィンドウターミナル16がディスプレイに表示され、デバッグ5の出力はウィンドウターミナル15に出力される。これにより、ユーザはウィンドウを切替えることなく、デバッグ作業を続けることができる。

【0016】ユーザ1が、図7のようなキャラクタ端末22を用いてデバッグを行っていた場合は、キャラクタ端末22から本装置を起動することによりデバッグ管理装置デバッグ管理装置の入力画面がキャラクタ端末22のディスプレイに表示される。以降、ユーザ1はキャラクタ端末22からデバッグ管理装置9に対してデバッグコマンドを送る。ユーザ1が、デバッグ管理装置9にデバッグ5の出力を表示するコマンドを送ると、デバッグ5の出力はキャラクタ端末22に出力される。同様に

6

して、ユーザ1が、デバッグ管理装置9にデバッグ7の出力を表示するコマンドを送ると、デバッグ5の出力はキャラクタ端末22に出力される。これにより、1台のキャラクタ端末22からデバッグ作業を続けることができる。

【0017】以上のように、この実施例は、ネットワークに接続された複数の計算機上で動作し、お互いに通信を行ないながら処理をする分散プログラムのデバッグにおいて、ユーザから送信先を示す識別子とデバッグコマンドを入力し、識別子から送信先のデバッグを判定し、遠隔もしくはローカルの計算機のデバッグ制御装置にデバッグコマンドを伝達するデバッグ管理装置と、デバッグ管理装置から送られるデバッグコマンドを入力しデバッグに出力する、また、デバッグの出力を遠隔もしくはローカルの計算に存在するデバッグ管理装置に伝達するデバッグ制御装置と、デバッグ管理装置が受信した各デバッグからの出力をデバッグ毎に管理し蓄積する、また、デバッグ管理装置からの要求により蓄積したデバッグの出力をユーザへ出力するデバッグ情報蓄積装置とを備え、ユーザが入力したデバッグコマンドを複数のデバッグに分配配送することを特徴とした分散プログラムデバッグ装置を説明した。

【0018】この実施例で、デバッグ管理装置は、ユーザから入力された識別子を解釈しローカル計算機もしくは遠隔計算機のデバッグにデバッグコマンドを分配配送する。よって、キャラクタ端末を使用しているユーザはキャラクタ端末間を移動することなく、また、ウィンドウターミナルを使用しているユーザはウィンドウを切替えることなく、分散しているデバッグにデバッグコマンドを送信することを可能とする。また、分散しているデバッグの出力をデバッグ情報蓄積装置によってデバッグ毎に蓄積し、ユーザが入力したデバッグ出力表示要求と表示デバッグを示す識別子により、指定されたデバッグの出力をいつでも取り出せる。さらに、デバッグコマンドの送信先が複数でも可能なようにすることで、同時に複数のデバッグに対してコマンドを送ることを可能にした。この実施例における分散デバッグ装置は、デバッグ管理装置によりユーザがキーボードから手を離すことなく同時に複数のデバッグに対してデバッグコマンドを送ることができる。

【0019】実施例2. 図4は、発明の実施例2を示す図であり、1～12は、上記実施例1と同様である。13は、ユーザ1によって時刻指定で入力されたデバッグコマンドをローカルのデバッグ制御装置10から入力し、コマンド実行時刻が来たならば実行コマンドをローカルのデバッグ制御装置10へ出力するローカルの時間管理装置である。14は、ユーザ1によって時刻指定で入力されたデバッグコマンドを遠隔のデバッグ制御装置11から入力し、コマンド実行時刻が来たならば実行コマンドを遠隔のデバッグ制御装置11へ出力する遠隔の

時間管理装置である。

【0020】次に上記実施例2の動作を図3及び図4を参照しながら説明する。ローカルのデバグ制御装置10が実行時間を指定されたデバグコマンドを入力したならば、そのコマンドをローカルの時間管理装置13へ送る。ローカルの時間管理装置13は、コマンドの時刻情報を解釈しローカル計算機2のタイマーを参照することにより、何秒後にそのコマンドを実行するかを算出する。指定時刻になったならば、デバグコマンドをローカルのデバグ制御装置10へ出力する。ローカルのデバグ制御装置10は、デバグコマンドをローカルのデバグ5へ出力することにより、指定時刻にデバグコマンドを実行することが可能となる。

【0021】以上のように、この実施例では、ユーザが入力したデバグコマンドとその実行時刻を解釈し、指定された時刻にデバグコマンドをデバグへ送信することをデバグ制御装置へ通知する時間管理装置を付加することにより、ユーザが指定した時刻にデバグコマンドをデバグに実行させることを特徴とした分散プログラムデバグ装置を説明した。

【0022】また、ユーザがデバグコマンドとそのコマンドをデバグで実行させる時刻を入力したならば、時間管理装置が、実行時刻を解釈し指定された時刻にデバグコマンドをデバグへ送ることを特徴とした分散プログラムデバグ装置を説明した。また、時間管理装置により複数のデバグに対して同時刻にデバグコマンドを送ることが可能となった。

【0023】このように、各プログラムに対して時刻を指定してコマンドが送れることで、例えば、マルチクライアント・サーバモデルのような分散プログラムにおいて、サーバに対しほぼ同時刻に複数のクライアントから要求が来た時のサーバの振舞いを管理制御することが行なえる。

【0024】実施例3. 上記説明では、分散プログラムデバグに利用する場合について述べたが、デバグコマンドに限らず、コマンド待ちとなっている分散するプログラムへコマンドを送信する場合や、指定時刻にプログラムへコマンドを送ることにも利用できる。例として図5に示すように、ローカルのプログラム18、遠隔のプログラム19をローカルのデバグ制御装置10、11に制御させる。ユーザ1が、ローカルのプログラム18にコマンドを送る場合は、デバグ管理装置9に対してローカルのプログラム18の識別子と送信するコマンドを入力する。デバグ管理装置9はプログラムの識別子を解釈し、ローカルのプログラム18を制御しているローカルのデバグ制御装置10にコマンドを送る。ローカルのデバグ制御装置10は、入力したコマンドをローカルのプログラム18へ送信する。ユーザ1がデバグ管理装置9に遠隔のプログラム19の出力を表示するコマンドを送ると、デバグ管理装置9は、デバグ

情報蓄積装置12に蓄積されている遠隔のプログラム19の出力情報を出力する。ローカルのデバグ制御装置10が実行時間を指定されたコマンドを入力したならば、そのコマンドをローカルの時間管理装置13へ送る。また、ローカルのデバグ制御装置10が実行時間を指定されたコマンドを入力したならば、そのコマンドをローカルの時間管理装置13へ送る。ローカルの時間管理装置13は、コマンドの時刻情報を解釈しローカル計算機2のタイマーを参照することにより、何秒後にそのコマンドを実行するかを算出する。指定時刻になったならば、コマンドをローカルのデバグ制御装置10へ出力する。ローカルのデバグ制御装置10は、コマンドをローカルのプログラム18へ出力することにより、指定時刻にローカルのプログラム18はコマンドを入力することが可能となる。

【0025】

【発明の効果】この発明は、以上説明したように構成されているので、ユーザの入力したプログラムに対するコマンドが分配配送されるので、同時に複数のプログラムに対してコマンドが送れ、分散プログラムの操作性が向上するという効果が得られる。

【図面の簡単な説明】

【図1】この発明の実施例1の分散プログラムデバグ装置を示す構成図である。

【図2】この発明のウィンドウ環境を用いた入出力の例の図である。

【図3】この発明装置に対してユーザが入力するデバグコマンドの例の図である。

【図4】この発明の実施例2の分散プログラムデバグ装置を表す構成図である。

【図5】この発明の他の実施例を説明するための構成図である。

【図6】従来の分散プログラムデバグを示す構成図である。

【図7】従来の分散プログラムデバグを示す構成図である。

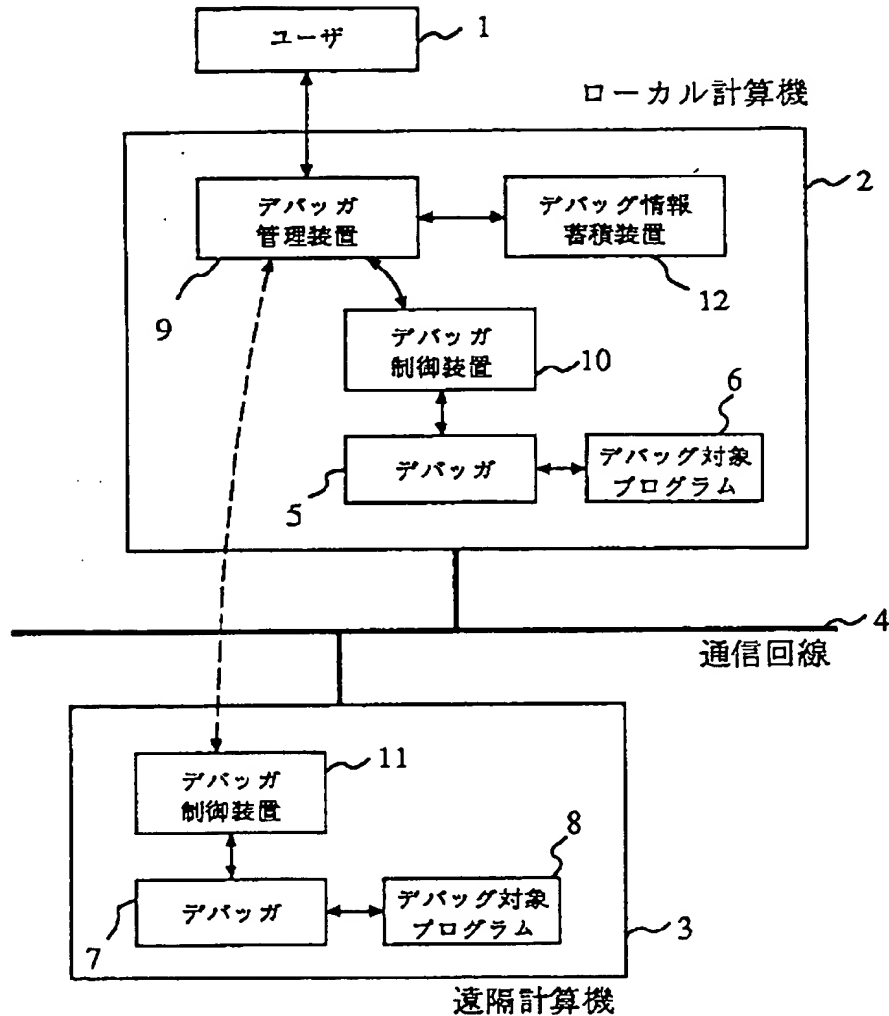
【符号の説明】

- 1 ユーザ
- 2 ローカル計算機
- 3 遠隔計算機
- 4 通信回線
- 5 ローカルのデバグ
- 6 デバグ対象プログラム
- 7 遠隔のデバグ
- 8 遠隔のデバグ対象プログラム
- 9 デバグ管理装置
- 10 ローカルのデバグ制御装置
- 11 遠隔のデバグ制御装置
- 12 デバグ情報蓄積装置
- 13 ローカルの時間管理装置

- 14 遠隔の時間管理装置
- 15 ローカルのデバッガのウィンドウ
- 16 遠隔のデバッガのウィンドウ
- 17 コマンド入力ウィンドウ
- 18 ローカルのプログラム
- 19 遠隔のプログラム

- 20 遠隔ログイン
- 21 ウィンドウターミナル
- 22 キャラクタ端末
- 23 遠隔デバッガのウィンドウターミナル
- 24 遠隔のデバッガのキャラクタ端末

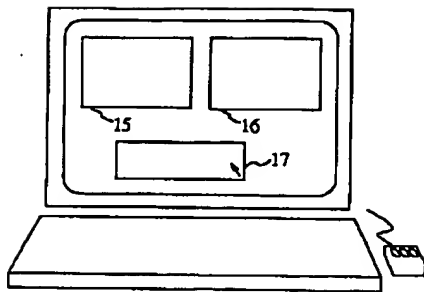
【図1】



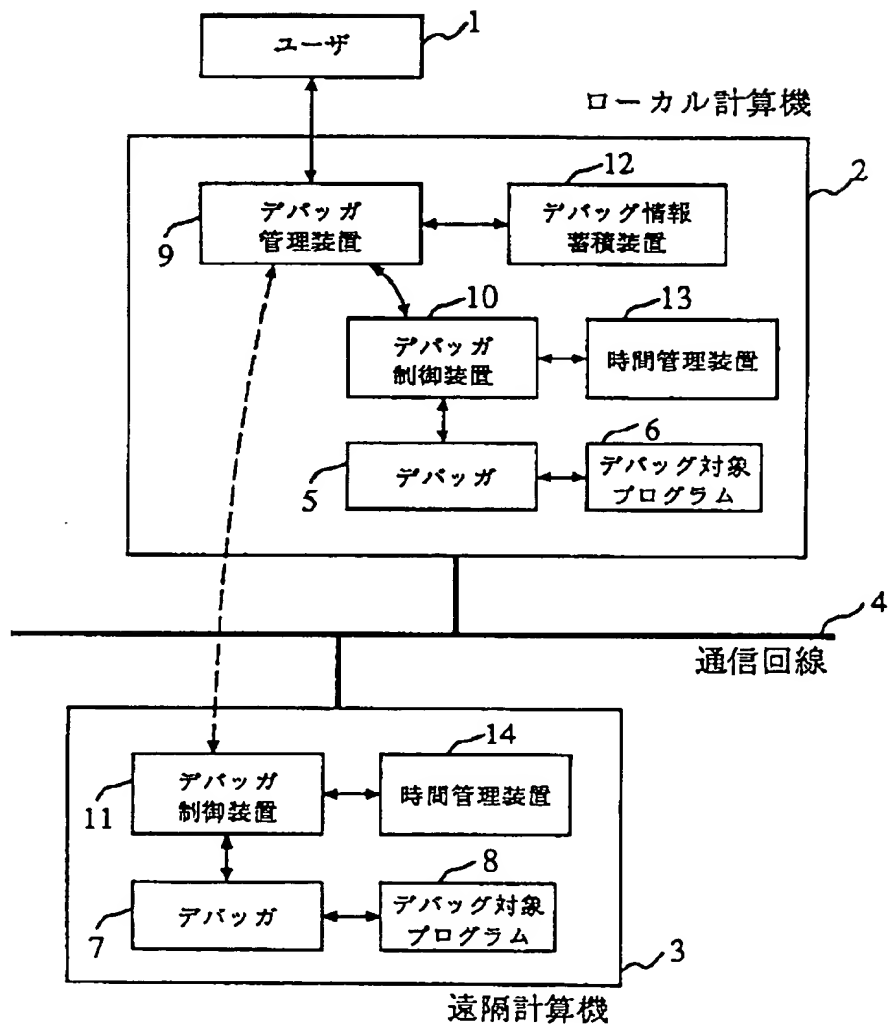
【図3】

- (a) debug ホスト名:プログラム名
- (b) :識別子1 :識別子2 デバッグコマンド
- (c) print :識別子1
- (d) :識別子1, 12:23:15 :識別子2, 12:23:15 デバッグコマンド
- (e) :識別子1, +20 :識別子2, +30 デバッグコマンド

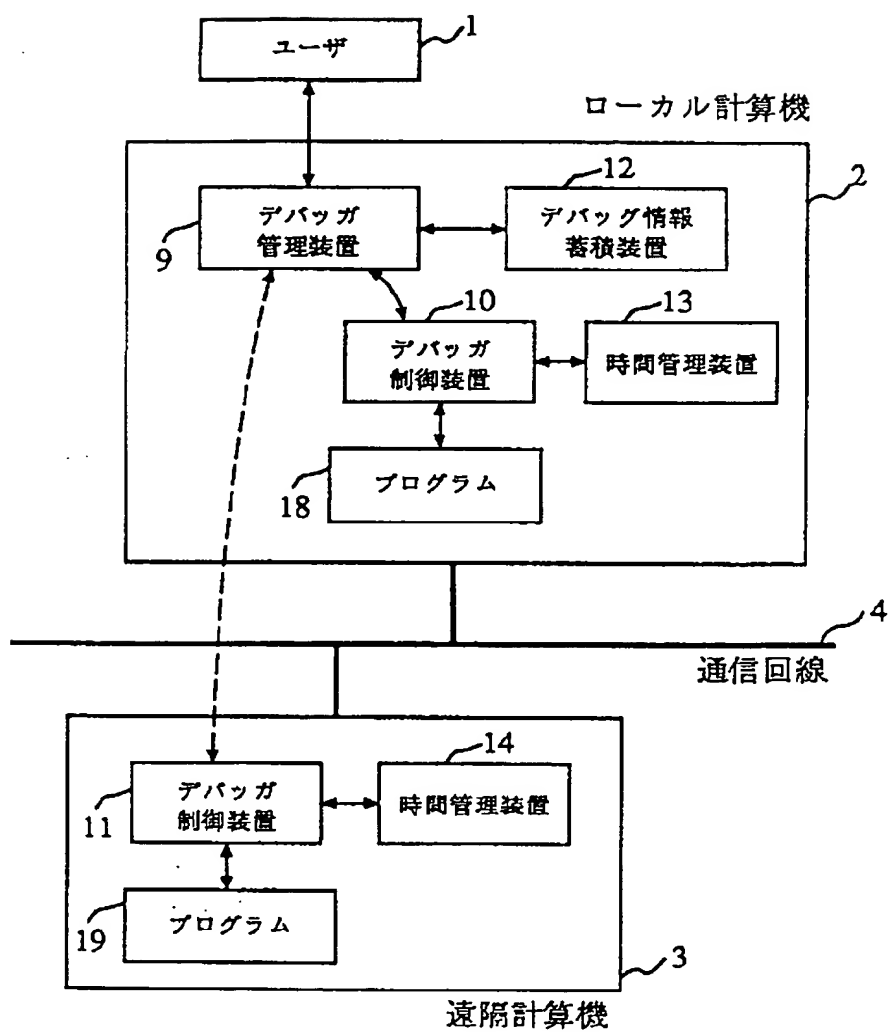
【図2】



【図4】

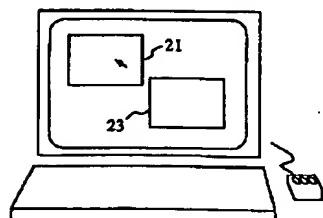


【図5】

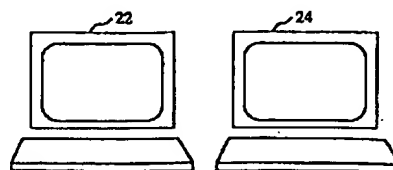


【図7】

ウィンドウ環境でデバッグを行なう場合



キャラクター端末を用いてデバッグを行なう場合



【図6】

